

[illegible]

3

Sy

MT
ME

MT

MT

MT

MT

MT

MT
MT

MT

MT
MT

MT
MT

MT
MT

MT
MT

MY
MT

MY
MT

MT
MTMT
MTMT
MTMT
MT

MT

MT
MT

MT

MT

MT

M1

W1
M1
22

M1

M1

1

1

1

1

1

```

LL          IIIIII          SSSSSSSS
LL          IIIIII          SSSSSSSS
LL          II             SS
LL          II             SS
LL          II             SS
LL          II             SS
LL          II             SSSSSS
LL          II             SSSSSS
LL          II             SS
LL          II             SS
LL          II             SS
LL          II             SS
LLLLLLLLLLLL IIIIII          SSSSSSSS
LLLLLLLLLLLL IIIIII          SSSSSSSS

```

(2) 50
(3) 76
(4) 113

HISTORY : Detailed Current Edit History
DECLARATIONS : Declarative Part of Module
MTH\$DTANH - Standard DOUBLE Precision Floating DTANH

```
0000 1 .TITLE MTH$DTANH ; Floating Point Hyperbolic Tangent routine
0000 2 ; (DTANH)
0000 3 .IDENT /1-011/ ; File: MTHDTANH.MAR Edit: JCW1011
0000 4 :
0000 5 :*****
0000 6 :
0000 7 :* COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0000 8 :* DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0000 9 :* ALL RIGHTS RESERVED.
0000 10 :
0000 11 :* THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0000 12 :* ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0000 13 :* INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0000 14 :* COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0000 15 :* OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0000 16 :* TRANSFERRED.
0000 17 :
0000 18 :* THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0000 19 :* AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0000 20 :* CORPORATION.
0000 21 :
0000 22 :* DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0000 23 :* SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0000 24 :
0000 25 :
0000 26 :*****
0000 27 :
0000 28 :
0000 29 : FACILITY: MATH LIBRARY
0000 30 :++
0000 31 : ABSTRACT:
0000 32 :
0000 33 : MTH$DTANH is a function which returns the floating point hyperbolic tangent
0000 34 : of its single precision floating point argument. The call is standard
0000 35 : call-by-reference.
0000 36 :
0000 37 :--
0000 38 :
0000 39 : VERSION: 01
0000 40 :
0000 41 : HISTORY:
0000 42 : AUTHOR:
0000 43 : Peter Yuo, 29-Jun-77: Version 01
0000 44 :
0000 45 : MODIFIED BY:
0000 46 :
0000 47 :
0000 48 :
```



```
0000 50      .SBTTL HISTORY ; Detailed Current Edit History
0000 51
0000 52
0000 53 : ALGORITHMIC DIFFERENCES FROM FP-11/C ROUTINE: none
0000 54 :
0000 55 : Edit History for Version 01 of MTH$DTANH
0000 56 :
0000 57 : 0-3  Call MTH$DCOSH, MTH$DSINH directly instead of using POLY.
0000 58 : 0-4  - Add .EXTRNs.  TNH 13-June-78
0000 59 : 0-5  - Use W offset on externals.  TNH 13-June-78
0000 60 :      More accurate.  TNH 13-June-78
0000 61 : 1-006 - Update version number and copyright notice.  JBS 16-NOV-78
0000 62 : 1-007 - Add " " to the PSECT directive.  JBS 22-DEC-78
0000 63 : 1-008 - Use MTH$DEXP_R6.  SBL 27-Sept-1979
0000 64 : 1-009 - Change constant 16.0 to 22.0 to correct inaccuracy.  The
0000 65 :      value of X above which 1.0 is the best machine approximation
0000 66 :      to DTANH(X) is about 20.10.  The next higher number that can
0000 67 :      be represented as a short literal is 22.0.  JAW 19-Sep-80
0000 68 : 1-010 - Use general mode addressing.  SBL 30-Nov-1981
0000 69 : 1-011 - Changed the constant 2^-14 to 2^-28 to correct inaccuracy.  For
0000 70 :      values of |X| between 2^-14 and 2^-28 DTANH was only accurate
0000 71 :      to 8 decimal places since the assumption that DTANH(x)=x in that
0000 72 :      range of values is false.  DTANH(X)=X for |X|<=2^-28 and
0000 73 :      DTANH(X)=DSINH(X)/DCOSH(X) for 2^-28<|X|<=.25.  All appropriate
0000 74 :      references to 2^-14 have been changed to 2^-28.  JCW 10-Jan-1983
```

```
0000 76      .SBTTL DECLARATIONS      ; Declarative Part of Module
0000 77
0000 78      ;
0000 79      ; INCLUDE FILES:
0000 80      ;
0000 81      ; EXTERNAL SYMBOLS:      MTH$JACKET_HDLR
0000 82      ;
0000 83      .DSABL GBL                ; Force .EXTRN on all symbols
0000 84      .EXTRN MTH$DCOSH          ; DCOSH
0000 85      .EXTRN MTH$DSINH          ; DSINH
0000 86      .EXTRN MTH$DEXP_R6        ; EXP
0000 87      ; EQUATED SYMBOLS:
0000 88
0000 89      SD_1.0 = ^F1.0            ; 1.0
0000 90      SD_22.0 = ^F22.0         ; 22.0
0000 91      value = 4                 ; value.rd.r
0000 92
0000 93      ;
0000 94      ; MACROS:      none
0000 95      ;
0000 96      ; PSECT DECLARATIONS:
0000 97
0000 98      .PSECT _MTH$CODE          PIC,SHR,LONG,EXE,NOWRT
0000 99      ; program section for math routines
0000 100     ;
0000 101     ; OWN STORAGE: none
0000 102     ;
0000 103     ;
0000 104     ; CONSTANTS:
0000 105     ;
0000 106     ;
0000 107     D_0.25:
0000 108     .WORD ^X3F80, 0, 0, 0      ; 0.25
0000 109     D_2_POWER_M28:
0000 110     .WORD ^X3280, 0, 0, 0      ; 2**-28
0010 111
```

```
0010 113 .SBTTL MTH$DTANH - Standard DOUBLE Precision Floating DTANH
0010 114
0010 115
0010 116 :++
0010 117 : FUNCTIONAL DESCRIPTION:
0010 118 :
0010 119 : DTANH - double precision floating point function
0010 120 :
0010 121 : DTANH(X) is computed as:
0010 122 :
0010 123 : If |X| <= 2**28, then DTANH(X) = X.
0010 124 : If 2**28 < |X| <= 0.25, then DTANH(X) = DSINH(X)/DCOSH(X).
0010 125 : If 0.25 < |X| < 22.0, then DTANH(X) = (DEXP(2*X) - 1) / (DEXP(2*X) + 1)
0010 126 : If 22.0 <= |X|, then DTANH(X) = sign(X) * 1
0010 127 :
0010 128 : CALLING SEQUENCE:
0010 129 :
0010 130 : DTANH.wd.v = MTH$DTANH(x.rd.r)
0010 131 :
0010 132 : INPUT PARAMETERS:
0010 133 :
00000004 0010 134 : LONG = 4 ; define longword multiplier
00000004 0010 135 : x = 1 * LONG ; Contents of x is the argument
0010 136 :
0010 137 : IMPLICIT INPUTS: none
0010 138 :
0010 139 : OUTPUT PARAMETERS:
0010 140 :
0010 141 : VALUE: double precision floating hyperbolic tangent of the argument
0010 142 :
0010 143 : IMPLICIT OUTPUTS: none
0010 144 :
0010 145 : COMPLETION CODES: none
0010 146 :
0010 147 : SIDE EFFECTS: none
0010 148 :
0010 149 : NOTE: This procedure disables floating point underflow, enables integer
0010 150 : overflow.
0010 151 :
0010 152 : ---
0010 153 :
0010 154 :
407C 0010 155 .ENTRY MTH$DTANH, ^M<IV, R2, R3, R4, R5, R6>
0012 156 : standard call-by-reference entry
0012 157 : disable DV (and FU), enable IV
0012 158 MTH$FLAG_JACKET : flag that this is a jacket procedure in
0012 :
6D 00700000'GF 9E 0012 MOVAB G^MTH$$JACKET_HND, (FP)
0019 : set handler address to jacket
0019 : handler
0019 159 : case of an error in routine
0019 160 : If an error, convert signal to user PC
0019 161 : and resignal
50 04 BC 70 0019 162 MOVD @value(AP), R0 : R0/R1 = |X| = @value(AP)
50 8000 BF AA 001D 163 BICW #^X8000, R0 : R0/R1 = |X|
E2 AF 50 71 0022 164 CMPD R0, D_2_POWER_M28 : compare |X| with 2**28
```



```

43 15 0026 165 BLEQ OUT_X ; branch if !X! <= 2**-28
      0028 166
      0028 167
      0028 168 ; 2**-28 < !X!
      0028 169
      0028 170
2B 50 71 0028 171 CMPD R0, S^#SD_22.0 ; compare !X! with 22.0
32 18 002B 172 BGEQ GEQ_TO_22.0 ; branch if !X! >= 22.0
      002D 173
      002D 174 ; 2**-28 < !X! < 22.0
      002D 175
      002D 176
      002D 177
CF AF 50 71 002D 178 CMPD R0, D 0.25 ; compare !X! with 0.25
17 15 0031 179 BLEQ LEQ_TO_0.25 ; branch if !X! <= 0.25
      0033 180
      0033 181
      0033 182 ; 0.25 < !X! < 22.0
      0033 183
      0033 184
50 04 BC 04 BC 61 0033 185 ADDD3 @value(AP), @value(AP), R0
      0039 186 ; R0/R1 = 2*X
      0039 187 JSB G^MTH$DEXP_R6 ; R0/R1 = DEXP(2*X)
52 50 08 61 003F 188 ADDD3 S^#SD_1.0, R0, R2 ; R2/R3 = DEXP(2*X) + 1
50 08 62 0043 189 SUBD S^#SD_1.0, R0 ; R0/R1 = DEXP(2*X) - 1
50 52 66 0046 190 DIVD R2, R0 ; R0/R1 = (DEXP(2*X) - 1) / (DEXP(2*X) + 1)
04 0049 191 RET ; return with result in R0/R1
      004A 192
      004A 193
      004A 194 ; 2**-1R6 < !X! <= 0.25
      004A 195
      004A 196
      004A 197 LEQ_TO_0.25:
00000000'GF 6C FA 004A 198 CALLG (AP), G^MTH$DCOSH ; R0/R1 = DCOSH(X)
52 50 70 0051 199 MOVD R0, R2 ; R2/R3 = DCOSH(X)
00000000'GF 6C FA 0054 200 CALLG (AP), G^MTH$DSINH ; R0/R1 = DSINH(X)
50 52 66 005B 201 DIVD R2, R0 ; R0/R1 = DSINH(X) / DCOSH(X)
04 005E 202 RET ; return with result in R0
      005F 203
      005F 204
      005F 205 ; !X! >= 22.0
      005F 206
      005F 207
      005F 208 GEQ_TO_22.0:
50 08 70 005F 209 MOVD S^#SD_1.0, R0 ; R0/R1 = 1.0
04 BC 73 0062 210 TSTD @value(AP) ; test the sign of X
50 03 18 0065 211 BGEQ 10$ ; branch if X >= 0
50 50 72 0067 212 MNEGD R0, R0 ; R0/R1 = -1
04 006A 213 10$: RET ; return with result in R0
      006B 214
      006B 215
      006B 216 ; !X! <= 2**-28
      006B 217
      006B 218
50 04 BC 70 006B 219 OUT_X: MOVD @value(AP), R0 ; R0/R1 = DTANH(X) = X
04 006F 220 RET ; return with result in R0/R1
0070 221
```


MTHSDTANH
1-011

; Floating Point Hyperbolic Tangent rout 16-SEP-1984 01:23:08 VAX/VMS Macro V04-00
MTHSDTANH - Standard DOUBLE Precision F 6-SEP-1984 11:22:57 [MTHRTL.SRC]MTHDTANH.MAR;1

Page 6
(4)

0070 222
0070 223
0070 224

.END

MTH\$DTANH
Symbol table

N 2
; Floating Point Hyperbolic Tangent rout 16-SEP-1984 01:23:08 VAX/VMS Macro V04-00
6-SEP-1984 11:22:57 [MTHRTL.SRC]MTHDTANH.MAR;1

Page 7
(4)

D_0.25	00000000	R	01
D_2_POWER_M28	00000008	R	01
GEQ_TO_22.0	0000005F	R	01
LEQ_TO_0.25	0000004A	R	01
LONG	= 00000004		
MTH\$\$JACKET_HND	*****	X	01
MTH\$DCOSH	*****	X	00
MTH\$DEXP_R6	*****	X	00
MTH\$DSINR	*****	X	00
MTH\$DTANH	00000010	RG	01
OUT_X	0000006B	R	01
SD_T.0	= 00004080		
SD_22.0	= 000042B0		
VALUE	= 00000004		

+-----+
! Psect synopsis !
+-----+

PSECT name	Allocation	PSECT No.	Attributes														
ABS	00000000 (0.)	00 (0.)	NOPIC	USR	CON	ABS	LCL	NOSHR	NOEXE	NORD	NOWRT	NOVEC	BYTE				
MTH\$CODE	00000070 (112.)	01 (1.)	PIC	USR	CON	REL	LCL	SHR	EXE	RD	NOWRT	NOVEC	LONG				

+-----+
! Performance indicators !
+-----+

Phase	Page faults	CPU Time	Elapsed Time
Initialization	29	00:00:00.08	00:00:00.70
Command processing	124	00:00:00.74	00:00:05.34
Pass 1	83	00:00:00.78	00:00:02.13
Symbol table sort	0	00:00:00.01	00:00:00.01
Pass 2	53	00:00:00.58	00:00:01.77
Symbol table output	3	00:00:00.02	00:00:00.06
Psect synopsis output	1	00:00:00.03	00:00:00.18
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	295	00:00:02.26	00:00:10.21

The working set limit was 900 pages.
3389 bytes (7 pages) of virtual memory were used to buffer the intermediate code.
There were 10 pages of symbol table space allocated to hold 15 non-local and 1 local symbols.
284 source lines were read in Pass 1, producing 11 object records in Pass 2.
1 page of virtual memory was used to define 1 macro.

+-----+
! Macro library statistics !
+-----+

Macro library name	Macros defined
_\$255\$DUA28:[SYSLIB]STARLET.MLB;2	0

0 GETS were required to define 0 macros.

There were no errors, warnings or information messages.

MTHSDTANH
VAX-11 Macro Run Statistics

B 3
; Floating Point Hyperbolic Tangent rout 16-SEP-1984 01:23:08 VAX/VMS Macro V04-00 Page 8
6-SEP-1984 11:22:57 [MTHRTL.SRC]MTHDTANH.MAR;1 (4)

MACRO/ENABLE=SUPPRESSION/DISABLE=(GLOBAL,TRACEBACK)/LIS=LISS:MTHDTANH/OBJ=OBJ\$:MTHDTANH MSRC\$:MTHJACKET/UPDATE=(ENHS:MTHJACKET)+MSRC

0260

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY